

What is claimed is:

1. A method of evaluating a set of memory maps for a program comprising a plurality of functions, the method comprising:
 - (a) executing a first version of the program according to a first memory map to generate a program counter trace;
 - (b) converting the program counter trace into a format defining a memory location in association with a function and an offset within the function using the first memory map;
 - (c) translating the program counter trace into physical addresses using one of the set of memory maps to be evaluated, different from the first memory map;
 - (d) evaluating the number of likely cache misses using a model of a direct-mapped cache for that one memory map; and
 - repeating steps (c) and (d) for each of the memory maps in the set.
2. A method according to claim 1, wherein step (c) is carried out by utilising the base address of each function of said one of the memory maps to be evaluated with the offset given in the program count trace format.
3. A method according to claim 1 or 2, wherein the direct-mapped cache model of step (d) emulates the operation of a cache which would occur if a new version of the program linked according to said one memory map under evaluation were to be executed.
4. A method according to any preceding claim, which requires the additional step of, subsequent to evaluating the first set of memory maps, generating a further set of memory maps for evaluation.
5. A method of operating a computer to evaluate a set of memory maps for a program comprising a plurality of functions, the method comprising:

loading a first version of the program into the computer and executing said first version to generate a program counter trace;

loading into the computer a memory map evaluation tool which carries out the steps of:

converting the program counter trace into a format defining a memory location in association with a function and an offset within the function using the first memory map;

translating the program counter trace into physical addresses using one of the set of memory maps to be evaluated, different from the first memory map; and

evaluating the number of likely cache misses using a model of a direct-mapped cache for that one memory map;

wherein the step of translating a program counter trace and evaluating the number of likely cache misses is repeated for each of the memory maps in a set to be evaluated.

6. A method according to claim 5, wherein the memory map generation tool is also operable to generate a further set of memory maps for evaluation taking into account the results of evaluation of the first set of memory maps.

7. A memory map evaluation tool comprising:

a first component operable to generate a program counter trace from execution of a first version of a program according to a first memory map and to provide from that program counter trace a converted format defining a memory location in association with a function and an offset within the function using the first memory map; and

a second component operable to translate the program counter trace into physical addresses using one of the set of memory maps to be evaluated, different from the first memory map, and to evaluate the number of likely cache misses using a model of a direct-mapped cache for that one memory map under evaluation.

8. A tool according to claim 7, in the form of program code means which, when executed on a computer, carry out the method steps of claim 1.

1007234.000000